
***Full credit is given to the above companies including the OS that this PDF file was generated!***

## *Linux Ubuntu 26.04 Manual Pages on command 'systemd-detect-virt.1'*

***$ man systemd-detect-virt.1***

SYSTEMD-DETECT-VIRT(1) systemd-detect-virt SYSTEMD-DETECT-VIRT(1)

NAME

systemd-detect-virt - Detect execution in a virtualized environment

SYNOPSIS

systemd-detect-virt [OPTIONS...]

DESCRIPTION

systemd-detect-virt detects execution in a virtualized environment. It identifies the virtualization technology and can distinguish full machine virtualization from container virtualization. systemd-detect-virt exits with a return value of 0 (success) if a virtualization technology is detected, and non-zero (error) otherwise. By default, any type of virtualization is detected, and the options --container and --vm can be used to limit what types of virtualization are detected.

When executed without --quiet will print a short identifier for the detected virtualization technology. The following technologies are currently identified:

Table 1. Known virtualization technologies (both VM, i.e. full hardware virtualization, and container, i.e. shared kernel virtualization)

```
??????????????????????????????????????????????????????????
? Type     ? ID          ? Product                 ?
??????????????????????????????????????????????????????????
? VM       ? qemu        ? QEMU software           ?
?          ?             ? virtualization, without ?
?          ?             ? KVM                     ?
?          ??????????????????????????????????????????????
```

? ? kvm ? Linux KVM kernel virtual ?

? ? ? machine, in combination ?

? ? ? with QEMU. Not used for ?

? ? ? other virtualizers using ?

? ? ? the KVM interfaces, such ?

? ? ? as Oracle VirtualBox or ?

? ? ? Amazon EC2 Nitro, see ?

? ? ? below. ?

? ??????????????????????????????????????????

? ? amazon ? Amazon EC2 Nitro using ?

? ? ? Linux KVM ?

? ??????????????????????????????????????????

? ? zvm ? s390 z/VM ?

? ??????????????????????????????????????????

? ? vmware ? VMware Workstation or ?

? ? ? Server, and related ?

? ? ? products ?

? ??????????????????????????????????????????

? ? microsoft ? Hyper-V, also known as ?

? ? ? Viridian or Windows ?

? ? ? Server Virtualization ?

? ??????????????????????????????????????????

? ? oracle ? Oracle VM VirtualBox ?

? ? ? (historically marketed by ?

? ? ? innotek and Sun ?

? ? ? Microsystems), for legacy ?

? ? ? and KVM hypervisor ?

? ??????????????????????????????????????????

? ? powervm ? IBM PowerVM hypervisor ? ?

? ? ? comes as firmware with ?

? ? ? some IBM POWER servers ?

? ??????????????????????????????????????????

? ? xen ? Xen hypervisor (only ?

?  ?  ? domU, not dom0)  ?

?  ??????????????????????????????????????????

?  ? bochs  ? Bochs Emulator  ?

?  ??????????????????????????????????????????

?  ? uml  ? User-mode Linux  ?

?  ??????????????????????????????????????????

?  ? parallels  ? Parallels Desktop,  ?

?  ?  ? Parallels Server  ?

?  ??????????????????????????????????????????

?  ? bhyve  ? bhyve, FreeBSD hypervisor ?

?  ??????????????????????????????????????????

?  ? qnx  ? QNX hypervisor  ?

?  ??????????????????????????????????????????

?  ? acrn  ? ACRN hypervisor[1]  ?

?  ??????????????????????????????????????????

?  ? apple  ? Apple virtualization  ?

?  ?  ? framework[2]  ?

?  ??????????????????????????????????????????

?  ? sre  ? LMHS SRE hypervisor[3]  ?

?  ??????????????????????????????????????????

?  ? google  ? Google Compute Engine[4]  ?

??????????????????????????????????????????????????????

? Container ? openvz  ? OpenVZ/Virtuozzo  ?

?  ??????????????????????????????????????????

?  ? lxc  ? Linux container  ?

?  ?  ? implementation by LXC  ?

?  ??????????????????????????????????????????

?  ? lxc-libvirt  ? Linux container  ?

?  ?  ? implementation by libvirt ?

?  ??????????????????????????????????????????

?  ? systemd-nspawn ? systemd's minimal  ?

?  ?  ? container implementation, ?

?  ?  ? see systemd-nspawn(1)  ?

?  ??????????????????????????????????????????????

?  ? docker  ? Docker container manager  ?

?  ??????????????????????????????????????????????

?  ? podman  ? Podman[5] container  ?

?  ?  ? manager  ?

?  ??????????????????????????????????????????????

?  ? rkt  ? rkt app container runtime ?

?  ??????????????????????????????????????????????

?  ? wsl  ? Windows Subsystem for  ?

?  ?  ? Linux[6]  ?

?  ??????????????????????????????????????????????

?  ? proot  ? proot[7] userspace  ?

?  ?  ? chroot/bind mount  ?

?  ?  ? emulation  ?

?  ??????????????????????????????????????????????

?  ? pouch  ? Pouch[8] Container Engine ?

??????????????????????????????????????????????????

If multiple virtualization solutions are used, only the "innermost" is detected and identified. That means if both machine and container virtualization are used in conjunction, only the latter will be identified (unless --vm is passed).

Windows Subsystem for Linux is not a Linux container, but an environment for running Linux userspace applications on top of the Windows kernel using a Linux-compatible interface. WSL is categorized as a container for practical purposes. Multiple WSL environments share the same kernel and services should generally behave like when being run in a container.

When executed with --cvm, instead of printing the virtualization technology, it will display the confidential virtual machine technology, if any. The following technologies are currently identified:

Table 2. Known confidential virtualization technologies

??????????????????????????????????????????????

? Arch  ? ID  ? Technology  ?

??????????????????????????????????????????????

? x86_64 ? sev  ? AMD Secure Encrypted  ?

?  ?  ? Virtualization  ?

? ??????????????????????????????????????

? ? sev-es ? AMD Secure Encrypted ?

? ? ? Virtualization - ?

? ? ? Encrypted State ?

? ??????????????????????????????????????

? ? sev-snp ? AMD Secure Encrypted ?

? ? ? Virtualization - Secure ?

? ? ? Nested Paging ?

? ??????????????????????????????????????

? ? tdx ? Intel Trust Domain ?

? ? ? Extensions ?

????????????????????????????????????????????

? s390x ? protvirt ? IBM Protected ?

? ? ? Virtualization (Secure ?

? ? ? Execution) ?

????????????????????????????????????????????

? arm64 ? cca ? Arm Confidential Compute ?

? ? ? Architecture ?

????????????????????????????????????????????

OPTIONS

The following options are understood:

-c, --container

Only detects container virtualization (i.e. shared kernel virtualization).

-v, --vm

Only detects hardware virtualization.

-r, --chroot

Detect whether invoked in a chroot(2) environment. In this mode, no output is written, but the return value indicates whether the process was invoked in a chroot() environment or not.

Added in version 228.

--private-users

Detect whether invoked in a user namespace. In this mode, no output is written, but the return value indicates whether the process was invoked inside of a user namespace or

not. See user_namespaces(7) for more information.

Added in version 232.

--cvm

Detect whether invoked in a confidential virtual machine. The result of this detection may be used to disable features that should not be used in confidential VMs. It must not be used to release security sensitive information. The latter must only be released after attestation of the confidential environment.

Added in version 254.

-q, --quiet

Suppress output of the virtualization technology identifier.

--list

Output all currently known and detectable container and VM environments.

Added in version 239.

--list-cvm

Output all currently known and detectable confidential virtualization technologies.

Added in version 254.

-h, --help

Print a short help text and exit.

--version

Print a short version string and exit.

## EXIT STATUS

If a virtualization technology is detected, 0 is returned, a non-zero code otherwise.

## SEE ALSO

systemd(1), systemd-nspawn(1), chroot(2), namespaces(7)

## NOTES

1. ACRN hypervisor

   https://projectacrn.org

2. Apple virtualization framework

   https://developer.apple.com/documentation/virtualization

3. LMHS SRE hypervisor

   https://www.lockheedmartin.com/en-us/products/Hardened-Security-for-Intel-Processors.html

4. Google Compute Engine

   https://cloud.google.com/compute

5. Podman

   https://podman.io

6. Windows Subsystem for Linux

   https://docs.microsoft.com/en-us/windows/wsl/about

7. proot

   https://proot-me.github.io/

8. Pouch

   https://github.com/alibaba/pouch